Amendments to the Specification:

Please replace paragraph 0009 with the following amended paragraph:

[0009] In one aspect, the invention is directed to a method of selectively recovering nodes on a computer network. The network detects an exception condition and recovers only the nodes within the scope of the exception condition. By "scope of the exception condition" is meant any node that is ~~effected~~ affected by the exception condition such that input/output ("I/O") requests are not processed. The network issues I/O requests to nodes during recovery that are not within the scope of the exception condition.

Please replace paragraph 0036 with the following amended paragraph:

[0036] It is noted that conventionally, the terms "FID," "LUN" and "LUN signature" have been used by EMC Corporation and others with reference to different types of storage arrays commercially available from EMC Corporation. While the term FID, LUN and LUN signature is used herein for ease of reference and understanding in describing the system and method, it is noted that such terms ~~is~~ are not limited to the specific types of storage arrays used by EMC Corporation, and ~~is~~ are intended to encompass any such type storage array, providing similar functionality, as used in the industry to provide identification and allow configuring and administration of storage devices in a network, as will be readily apparent to those of ordinary skill in the art.

Please replace paragraph 0046 with the following amended paragraph:

[0046] There are two general types of events illustrated in Figure 3: recover Host Bus Adapter ("HBA") events, which indicate that recovery is needed on the adapters, and recover FID or LUN events, which indicate that recovery is needed on the LUNs or FIDs. Recovery of the FID is necessary in order to recover the LUNs associated with the FID. Therefore, both FID events and LUN events trigger the same subroutine to recover the FIDs. However, prior to determining if the event is an HBA event or a FID/LUN event, when the event occurs, it is first determined if it is a shutdown event 55. If yes, then the routine stops 57. If no, if ~~If~~ an HBA

event occurs at step 59, the operating system runs a program routine to recover the HBA flow at step 61. The program routine to recover HBA flow is illustrated in Figure 4 and discussed in more detail below. If a recover units event occurs at step 63, the operating system runs a program routine to recover the units at step 65. The program routine for recovering units is shown in Figure 5 and also discussed in detail below. If an exception event occurs that is not recognizable as a units or HBA event, then the error is not able to be fixed. The program routine then proceeds to step 67 where a "dump" of system information is stored for analysis. The entire system is stopped at step 69. The stored system information may then be analyzed by technicians to determine what error occurred.

Please replace paragraph 0055 with the following amended paragraph:

[0055] If a recover units event has been triggered, for example, at steps 95, 85, or 83 in Figure 34, the operating system will detect the units event at step 63 in Figure 3. The operating system will then run the recover FIDs routine ~~it~~in step 65. The recover FIDs routine is illustrated in Figure 5.

Please replace paragraph 0059 with the following amended paragraph:

[0059] At step 123, the program routine queries whether the current pointer is set at the end of the list. If the pointer is not at the end of the list, a program routine to recover the FID flow of one ~~fid~~FID is run from the operating system at step 125. The program routine at step 125 is illustrated in Figure 6 and discussed in greater detail below.

Please replace paragraph 0068 with the following amended paragraph:

[0068] If the FID login was not successful at step 193, the FID is marked at step 197 according to the cause of error. The possible causes of error are discussed in more detail in the discussion accompanying Figure 7. If the login was successful, the FID state is set to Fid_Ok at step 195, the routine is ~~not~~now finished with the FID at step 199, and continues to step ~~159~~201.

Please replace paragraph 0069 with the following amended paragraph:

[0069] If the FID does not need to be logged in at step 177, the routine queries if the FID state is Fid_Ok at step 179. If the FID state is "Ok," a program routine to recover the LUNs flow is run at step 183. This program routine is illustrated in Figure 8. As will understood from the discussion accompanying Figure 8 and 12, the program routine returns a status that the LUN recovery was either successful or unsuccessful, and these results are analyzed at step 185, which runs the program routine illustrated in Figure 12. If the FID state is not "Ok" at step 179, then the FID state is set to Fid_Dead at step 181. In either case, the routine is finished with the FID at step 199 and thus, at 201 returns to step 125 in Figure 5.

Please replace paragraph 0073 with the following amended paragraph:

[0073] If the driver has a record of the FID existing previously, the program routine queries how many login attempts ~~there~~ have been made at step 419. If there have been too many login attempts made at step 419, the FID state is set to Fid_Dead at step 423, and the state information is returned at step 429 to Figure 6 (step 191). By "too many login attempts" is meant a number of attempts indicating that future recovery is unlikely. The number of attempts may be determined experimentally in a particular system, but is preferably about five attempts that occur between two and four seconds apart.

Please replace paragraph 0078 with the following amended paragraph:

[0078] At step 227, the routine queries whether the return state is ~~Ok or~~ success. If the routine to recover the LUN was successful, the status of the LUN is set to Lun_Ok at step 229. At step 255, the pointer is advance to the next LUN on the list, and the program routine returns to step 215.

Please replace paragraph 0082 with the following amended paragraph:

[0082] Referring now to Figure 12, at step 450, the program routine receives the status and flags that were set in step 219 of Figure 8, which determines the steps performed in

Figure 12. In step 452, if the status was set to "Ok," indicating that the LUNs were successfully recovered, the program routine proceeds to step 462 where the FID state is set to Fid_Ok to indicate that the FID is available ~~o~~for any and all I/O requests. If the status is not "Ok," the program routine proceeds to step 454 where the status is checked to determine if a signature error occurred. If a signature error occurred, then the data on the FID and its associated LUNs cannot be validated. To prevent corruption, the program routine proceeds to step 464, where the FID state is set to Fid_Dead_Forever to prevent any attempts at future recovery.